

Volume 7, No.2 Agustus 2006

ISSN 1411 - 4453

Akreditasi :

SK No. : 23a / DIKTI / Kep / 2004

# **Pakar**

**JURNAL TEKNOLOGI INFORMASI DAN BISNIS**

**Membangun Sistem Berbasis Pengetahuan Pada Manajemen Terpadu Balita Sakit**

*Erik Iman Heri Ujjianto*

**Konversi Pemodelan Dalam UML Ke Skema Pemrograman Menggunakan Rational Rose**

*Febri Nova Lenti*

**Penerapan Metode Material Requirement Planning ( MRP ) Untuk Mendisain Sistem Informasi Perencanaan Dan Pengendalian Persediaan Bahan Bangunan**

*Dibyo Susilo*

**Analisis Dan Perancangan Sistem Pendaftaran Peserta Praktikum Laboratorium Komputer Berbasis Client/Server**

*Iwan hartadi Tri Untoro*

**Enkripsi SMS Dengan Chaos Based Stream Cipher**

*Susany Soplanit, Cliff Hilary Wibisono*

**Sistem Informasi Penjualan Bahan Bakar Pada SPBU Berbasis UML Dengan Menggunakan Rational Rose 2000**

*Suhirman*

**Sistem Penyimpanan Terpusat Pada Sistem Jaringan Komputer**

*Wagito*



Diterbitkan oleh :

**Pusat Pengembangan Sains dan Teknologi  
Fakultas Sains dan Teknologi  
Universitas Teknologi Yogyakarta**



# ***Pakar***

## **JURNAL TEKNOLOGI INFORMASI DAN BISNIS**

---

- |   |           |
|---|-----------|
| Membangun Sistem Berbasis Pengetahuan pada Manajemen Terpadu Balita Sakit<br><u>Erik Iman Heri Ujianto</u>  | 95 – 99   |
| Konversi Pemodelan Dalam UML Ke Skema Pemrograman Menggunakan <i>Rational Rose</i><br><u>Febri Nova Lenti</u>   | 101 – 115 |
| Penerapan Metode Material Requirement Planning ( MRP ) Untuk Mendisain Sistem Informasi Perencanaan Dan Pengendalian Persediaan Bahan Bangunan<br><u>Dibyo Susilo</u> | 117 – 132 |
| Analisis Dan Perancangan Sistem Pendaftaran Peserta Praktikum Laboratorium Komputer Berbasis Client/Server<br><u>Iwan Hartadi Tri Untoro</u>                          | 133 – 147 |
| Enkripsi SMS Dengan Chaos Based Stream Cipher<br><u>Susany Soplanit, Cliff Hilary Wibisono</u>  | 149 – 157 |
| Sistem Informasi Penjualan Bahan Bakar Pada SPBU Berbasis UML Dengan Menggunakan <i>Rational Rose</i> 2000<br><u>Suhirman</u>   | 159 – 168 |
| Sistem Penyimpanan Terpusat Pada Sistem Jaringan Komputer<br><u>Wagito</u>  | 169 – 180 |

## KONVERSI PEMODELAN DALAM UML KE SKEMA PEMROGRAMAN MENGGUNAKAN RATIONAL ROSE

*Oleh: Febri Nova Lenti*

### ABSTRACT

- *Forward Engineering* adalah salah satu fitur dari *Rational Rose*, suatu perangkat lunak yang memiliki perangkat-perangkat pemodelan secara visual untuk membangun suatu solusi dalam rekayasa perangkat lunak dan pemodelan bisnis. *Rational Rose* memakai UML sebagai bahasa pemodelannya.

Oleh karena UML bukan suatu metode, tetapi bahasa pemodelan, maka untuk menyelesaikan atau mengembangkan Sistem Informasi Wartel diterapkan metode berorientasi obyek *Coad yourdan* dengan menggunakan notasi UML memakai perangkat lunak bantu *Rational Rose*. Kemudian dilakukan *Forward Engineering* terhadap diagram kelas yang dimodelkan sehingga menghasilkan kode skema program dari masing-masing kelas dalam bahasa C++.

*Keywords : Forward Engineering, UML, diagram kelas, usecase*

### PENDAHULUAN

Menurut Hariyanto (2004) UML adalah bahasa standar pemodelan visual dalam rekayasa perangkat lunak, yang memberikan cara standar untuk menggambarkan cetak biru bagi perangkat lunak yang dibangun. Penggunaan UML berdampak pada peningkatan produktivitas dan kualitas, serta pengurangan biaya dan waktu.

Namun karena UML hanya merupakan bahasa untuk pemodelan maka UML bukanlah rujukan bagaimana melakukan analisis dan desain berorientasi obyek. Untuk mengetahui bagaimana melakukan analisis dan desain berorientasi obyek secara baik, dapat digunakan beberapa metodologi seperti Metode Booch, metode Coad and Yourdan, Metode Jacobson, Metode Rumbaugh, Metode Wirfs-Brock atau RUP. Pada penelitian ini digunakan metode Coad Yourdan.



Salah satu perangkat lunak bantu (*tool*) yang mengimplementasikan bahasa UML adalah *Rational Rose*. Salah satu kelebihan *Rational Rose* adalah mendukung *Forward Engineering* dan *Reverse Engineering*. *Forward Engineering* adalah kemampuan untuk mentransformasikan (*generate*) model yang dibuat menjadi kode dalam beberapa bahasa pemrograman. *Reverse Engineering* adalah kemampuan untuk menghasilkan gambaran arsitektur dari kode-kode perangkat lunak aplikasi.

Pada tulisan ini akan dibahas mengenai konversi model Sistem Informasi Wartel dalam notasi UML dengan menggunakan *Rational Rose*. Model yang dibuat adalah berbasis pada hasil analisis dan desain berorientasi object (OOA dan OOD) metode Coad and Yourdan.

## LANDASAN TEORI

Lenti (2001) Hasil-hasil analisis menggunakan metode Coad Yourdan, pada studi kasus WaIS adalah sebagai berikut :

WaIS memiliki kemampuan- kemampuan , yaitu :

- Menerima sinyal mulai suatu transaksi percakapan telepon (*SRS-WaIS-Fungsi Produk*)
- Menghitung biaya transaksi per menit berdasarkan *Tarif, Diskon, dan Airtime* (*SRS-WaIS-Fungsi Produk*)
- Mengupdate data di *TmpTrans* (*SRS-WaIS-Fungsi Produk*)
- Menerima sinyal selesai transaksi (*SRS-WaIS-Fungsi Produk*)
- Menghitung total biaya yang terdiri dari komponen biaya transaksi, pajak, dan biaya layanan (*SRS-WaIS-Fungsi Produk*)
- Mencetak total biaya ke printer (*SRS-WaIS-Fungsi Produk*)
- Menerima pembayaran transaksi (*SRS-WaIS-Fungsi Produk*)
- Menuliskan total biaya transaksi ke *Transaksi* (*SRS-WaIS-Fungsi Produk*)
- Menghapus data transaksi dari *TmpTrans* (*SRS-WaIS-Fungsi Produk*)
- Memproses login pengguna (*SRS-WaIS-Fungsi Produk*)
- Melakukan rekapitulasi *Transaksi* secara periodik (*SRS-WaIS-Fungsi Produk*)
- Melakukan pencetakan report *Transaksi* (*SRS-WaIS-Fungsi Produk*)
- Menghapus data dari *Transaksi* (me-reset *Transaksi*) (*SRS-WaIS-Fungsi Produk*)
- Melakukan backup data *Transaksi* (*SRS-WaIS-Fungsi Produk*)



Deksripsi Kelas-&-Objek yang dipergunakan terangkum pada tabel 1 berikut:

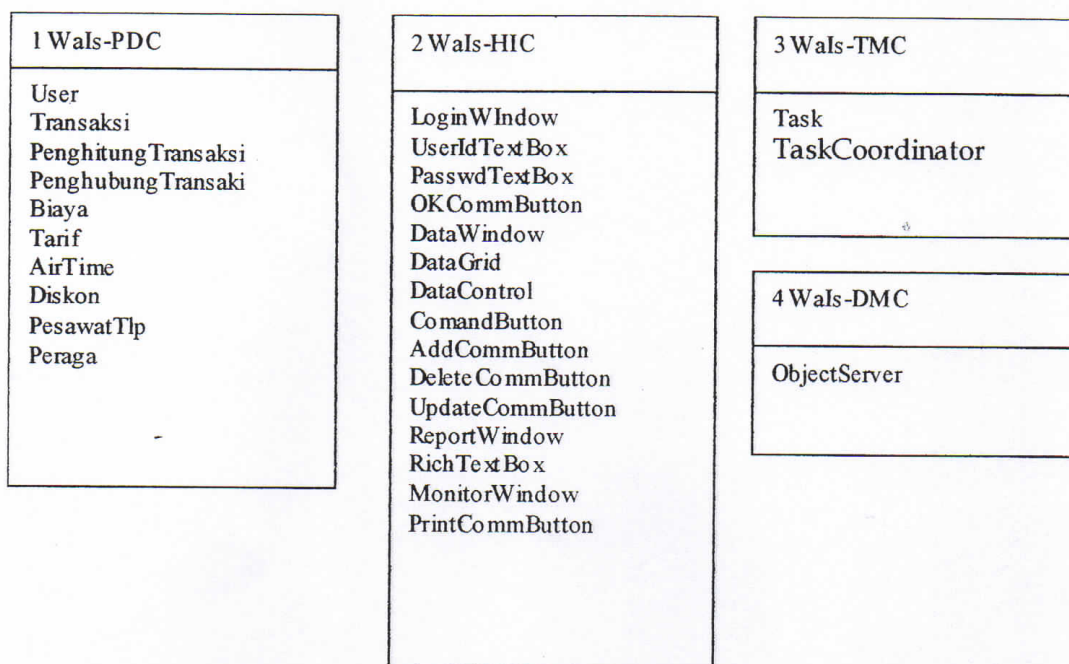
**Tabel 1.** Deskripsi Kelas-&-Objek

No	Nama	Alasan Pemilihan	Deskripsi	Nama Lain
1	User	<i>Role Played</i> - E	Pengguna Sistem terdiri dari Kasir dan Admin	Pengguna
2	Transaksi	<i>Operational Procedure</i> - F	Prosedur penggunaan pesawat telepon	Percakapan telepon
3	TmpTrans	<i>Operational Procedure</i> - F	Prosedur penghitungan biaya penggunaan pesawat telepon	Penghitung Trans
4	Penghubung Trans	<i>Operational Procedure</i> - F	Prosedur penghubung pesawat telepon dengan transaksi	
5	Tarif	<i>Things Remembered</i> - D	Tarif telepon	
6	AirTime	<i>Things Remembered</i> - D	Tarif airtime	
7	Diskon	<i>Things Remembered</i> - D	Diskon tarif	
8	PesawatTlp	<i>Device</i> - C	Pesawat telepon yang digunakan	Telepon
9	PapanPeraga	<i>Device</i> - C	Papan peraga telepon, menampilkan no tujuan, waktu dan biaya percakapan per pulsa	Display

#### Hasil Perancangan WaIS dengan Metode Coad Yourdan

Lenti (2001) Setelah keempat aktivitas perancangan dilakukan, gambar 1 berikut ini adalah gambaran hasil OOD:





**Gambar 1.** Hasil Perancangan Berorientasi Objek

## IMPLEMENTASI

### Pemodelan UML

Pada penelitian ini akan digunakan 2 diagram untuk pemodelan yaitu:

- 1) Diagram usecase (*Usecase diagram*) untuk menggambarkan perilaku sistem (aspek dinamis) dan
- 2) Diagram kelas (*Class diagram*) untuk melihat struktur sistem (aspek statis).

### Diagram Usecase

Metode OOA dan OOD Coud Yourdan tidak menggunakan usecase dalam analisisnya. Namun dari hasil analisis yang tercantum di dokumen SRS (*Software Requirement Specification*) dan hasil perancangan HIC (*Human Interface Component*) dengan beberapa modifikasi maka dapat disusun usecase yang dikategorikan berdasarkan pemakai/aktor sebagai berikut:



---

1. Aktor : Pelanggan

*Usecase :*

- a. Membuat sambungan telepon
- b. Memutuskan sambungan telepon
- c. Memonitor biaya selama percakapan
- d. Membayar biaya percakapan
- e. Menerima bon pembayaran

2. Aktor : Kasir

*Usecase :*

- a. Login ke sistem
- b. Memonitor Pelanggan (pelanggan yang tersambung dan terputus, biaya percakapan)
- c. Menerima pembayaran
- d. Mencetak bon pembayaran
- e. Memberikan bon pembayaran

3. Aktor : Admin

*Usecase :*

- a. Login
- b. Mengelola data user
- c. Mengelola data biaya
- d. Mengelola data transaksi

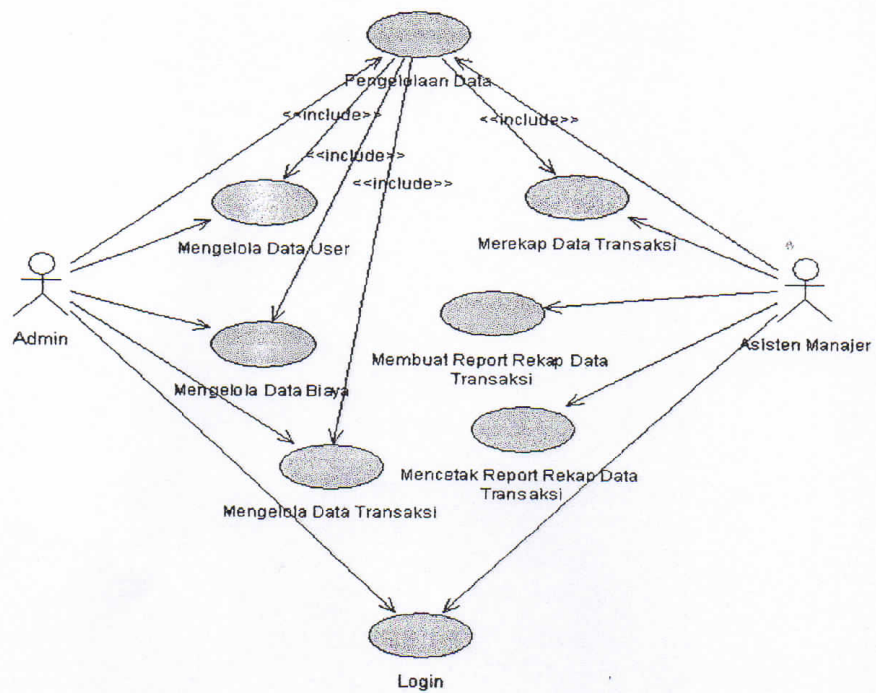
4. Aktor : Asisten Manajer

*Usecase :*

- a. Login
- b. Merekap data transaksi
- c. Membuat report rekap data transaksi
- d. Mencetak report rekap data transaksi

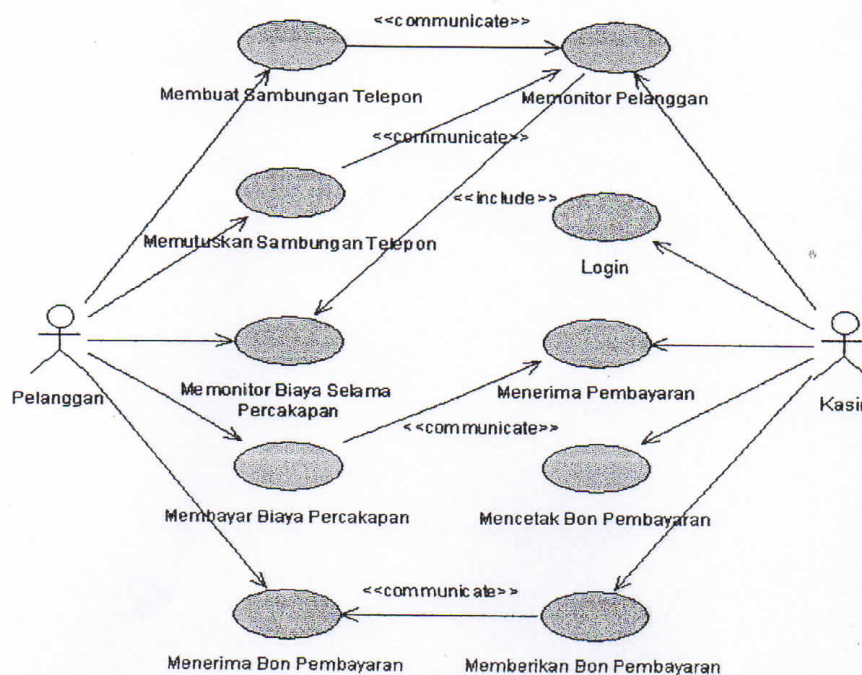
Beberapa *usecase* yang berkaitan dengan pengelolaan data (mengelola data user, mengelola data biaya, mengelola data transaksi dan merekap data transaksi) dapat di buat satu *usecase* induk yaitu *usecase* Pengelolaan data. Agar mudah dibaca, *usecase-usecase* tersebut dibuat pada 2 diagram terpisah yaitu Diagram *Usecase* Transaksi (gambar 2.) dan Diagram *Usecase* Administrasi Data (gambar 3.).





**Gambar 2.** Diagram *Usecase* Transaksi



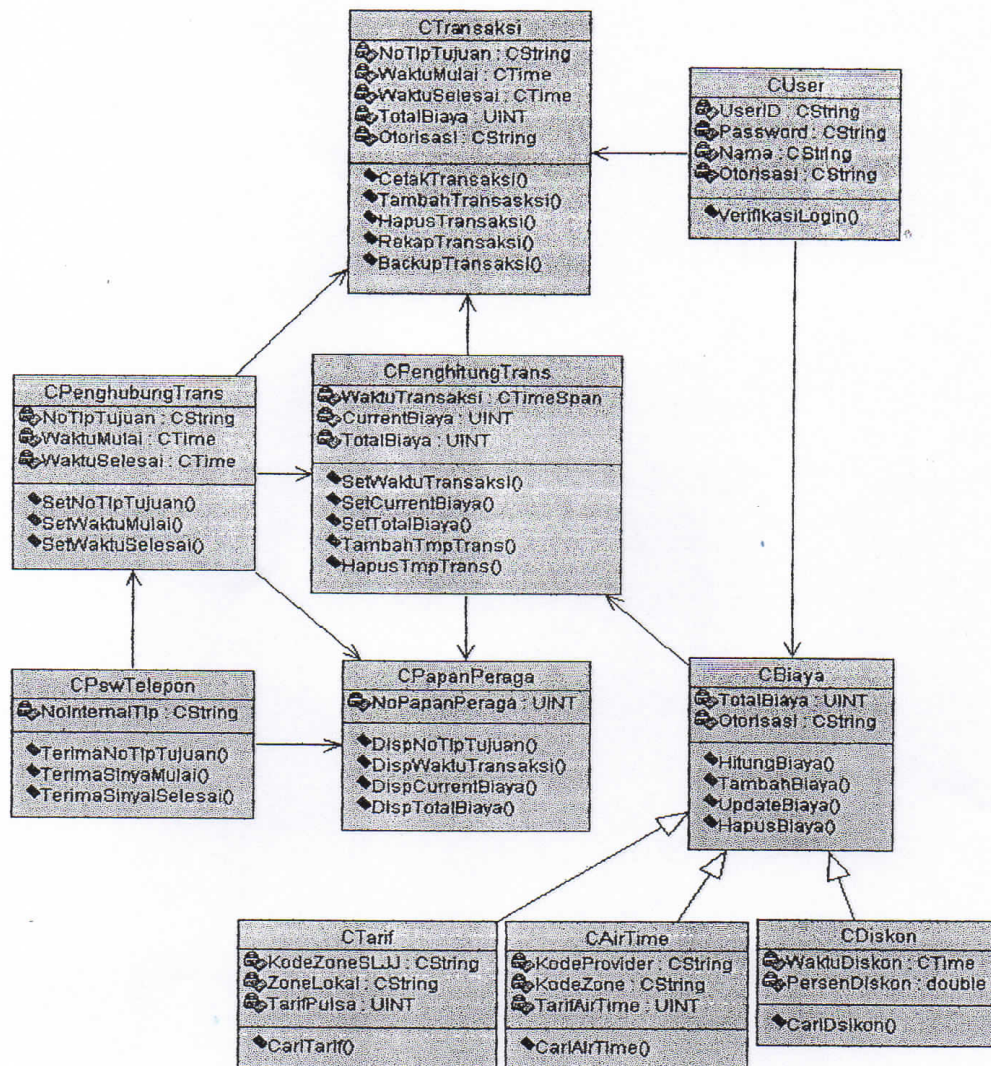


**Gambar 3.** Diagram *Usecase* Administrasi Data

#### Diagram kelas

Dari diagram kelas WaIS PDC berdasarkan hasil perancangan metode Coad Yourdan (gambar 1) dengan beberapa perubahan yang disesuaikan dengan sintaks bahasa pemrograman C++ yang ada di Microsoft Visual C++ maka dibuat diagram kelas seperti gambar 4 berikut:





**Gambar 4.** Diagram kelas WAIS

Beberapa perubahan model diagram kelas pada gambar 4. dibandingkan dengan perancangan PDC pada gambar 1. yaitu:

1. Nama kelas pada diagram kelas dibandingkan dengan desain PDC adalah penambahan huruf 'C' di awal nama kelas seperti kelas Transaksi menjadi CTransaksi. Ini dimaksudkan untuk menyesuaikan dengan lingkungan Microsoft Visual C++ dimana setiap nama kelas diawali dengan huruf 'C' untuk membedakannya dengan nama atribut dan operasi.

2. Tipe data atribut ditambahkan pada diagram kelas dengan menggunakan tipe data yang dikenal pada MS Visual C++. Ini dilakukan karena pada waktu proses forward engineering setiap atribut harus mempunyai tipe data agar bisa dikodekan. Kalau hal ini tidak dilakukan maka atribut pada model tidak akan 'ter-forward' menjadi kode.
3. Attribute TglTransaksi dan operasi SetTglTransaksi pada kelas Transaksi dan PenghitungTrans di PDC tidak digunakan lagi pada kelas CTransaksi dan kelas CPenghitungTrans di diagram kelas, karena tipe data object 'CTime' untuk atribut 'WaktuMulai' dan 'WaktuSelesai' sudah merepresentasikan tanggal (date) dan waktu (time).
4. Penambahan kopling (relasi) antara kelas CBiaya dan CUser. Seperti halnya CTransaksi, CBiaya juga membutuhkan data otorisasi user dari CUser untuk bisa melakukan operasi pengelolaan data (TambahBiaya(), UpdateBiaya() dan HapusBiaya()).

Diagram kelas yang dibuat tidak mencakup kelas-kelas yang ada pada desain HIC (*Human Interface Component*) dikarenakan lingkungan pengembangan perangkat lunak yang digunakan adalah bahasa pemrograman visual yaitu Microsoft Visual C++ yang memudahkan pengembang untuk membuat GUI dengan kelas-kelas yang tinggal pakai (*reusable*) seperti :

- CFormView ataupun CDialog untuk pembuatan form
- CEditView untuk membuat jendela editor teks.
- CButton untuk membuat kontrol tombol.
- CComboBox untuk membuat kontrol combo.
- CListCtrl untuk membuat kontrol list.
- CMenu untuk pembuatan menu.
- DLL.

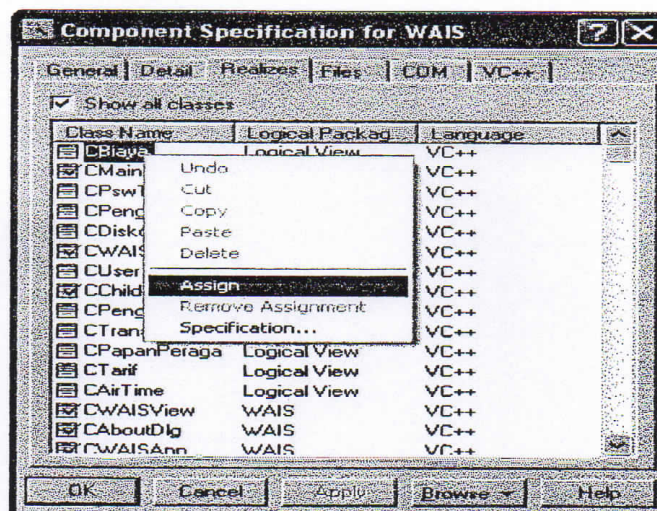
Diagram kelas tidak juga mencakup kelas-kelas yang ada pada desain TMC (*Task Management Component*) yang mengorganisir task (*multitasking*) dikarenakan pengembangan perangkat lunak dilakukan pada sistem operasi MS Windows yang mendukung *multitasking*. Untuk membangkitkan task MS Windows menggunakan Even driven (yang terpicu oleh penekanan keyboard, pergerakan dan penekanan tombol mouse, clock system, dll). Oleh karena itu seluruh bahasa pemrograman di lingkungan sistem operasi MS-Windows (Visual Basic, Visual C++, Borland C++, JavaBuilder, C++ Builder dll) sudah pasti mendukung even driven ini untuk menjalankan suatu task/prosedur. Sedangkan pengelolaan task-task tersebut sudah secara 'otomatis' oleh sistem operasi yang memang mendukung *multitasking*.



### Forward engineering

*Forward engineering* akan mengupdate kode pada project MSVC++ berdasarkan pemodelan WAIS yang telah dibuat. Langkah-langkah untuk melakukan *forward engineering* adalah sebagai berikut:

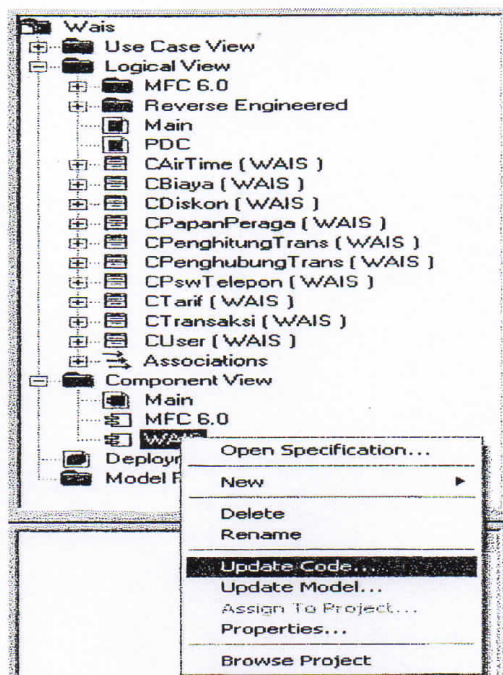
1. Assignment pada kelas-kelas yang dibuat pada pemodelan WAIS dengan cara double click pada component WAIS untuk membuka jendela '*Component Specification for WAIS*', kemudian klik tab '*Realizes*' dan klik kanan pada kelas-kelas yang akan di-assign seperti tampak pada gambar 5. berikut:



Gambar. 5. Jendela 'Component Specification for WAIS'

Kemudian pada popup menu, klik teks '*Assign*'. Lakukan hal tersebut untuk semua kelas yang telah dibuat pada pemodelan WAIS. Setelah selesai klik tombol '*OK*'.

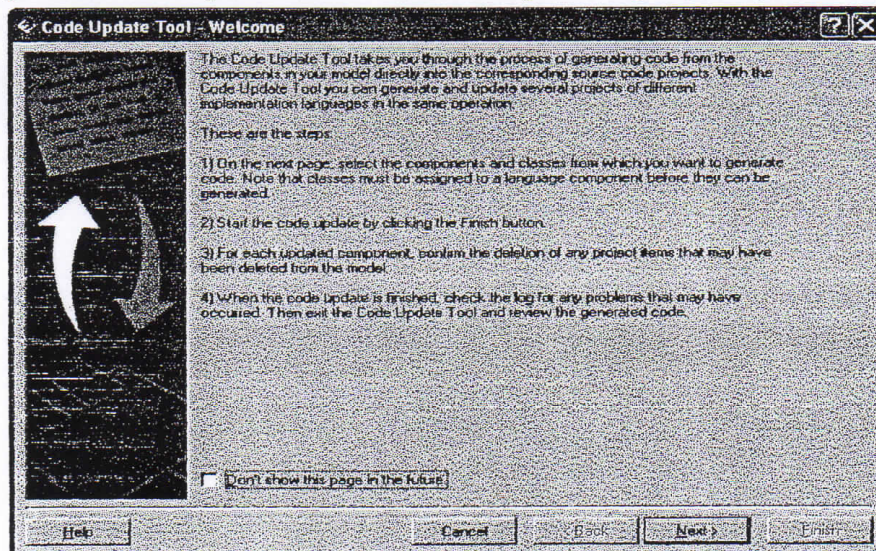
2. Klik kanan pada komponen WAIS di folder '*Component View*' maka akan muncul popup menu seperti tampak pada gambar. 6. berikut:



Gambar. 6. Pop-up Menu untuk update kode

Klik teks 'Update Code...'

3. Tampil jendela 'Code Update Tool – Welcome' seperti tampak pada gambar. 7 berikut:

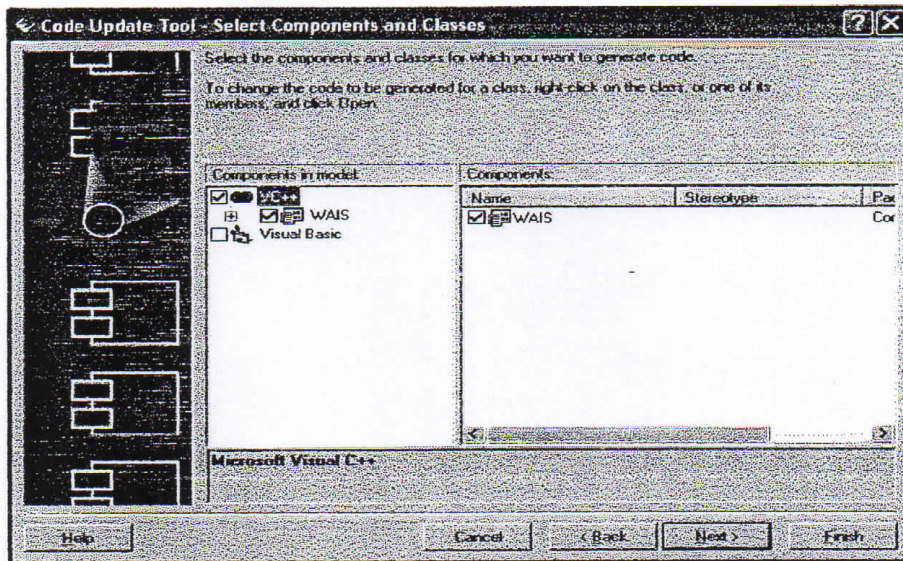




**Gambar. 7.** Jendela 'Code Update Tool – Welcome'

Klik tombol 'Next'.

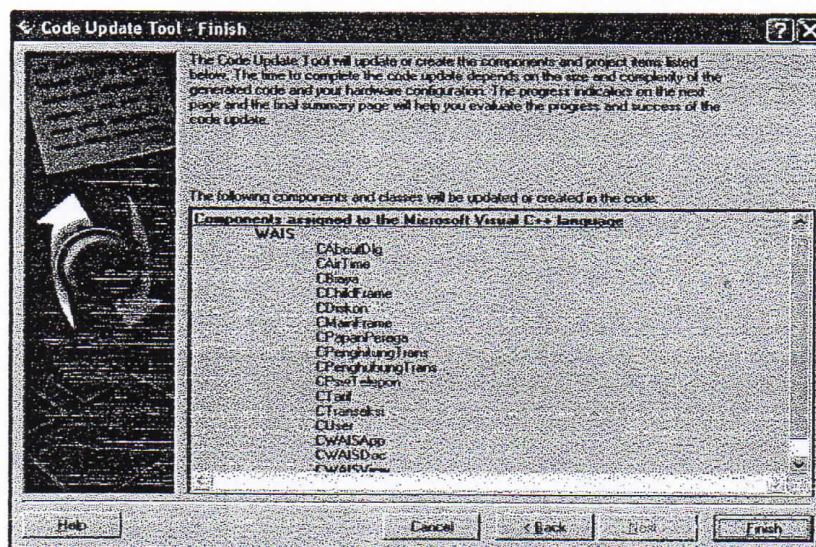
4. Tampil jendela 'Code Update Tool – Select Components and Classes' seperti tampak pada gambar. 8 berikut:

**Gambar. 8.** Jendela 'Code Update Tool – Select Components and Classes'

Pastikan checkbox pada teks 'WAIS' terset. Klik tombol 'Next'.

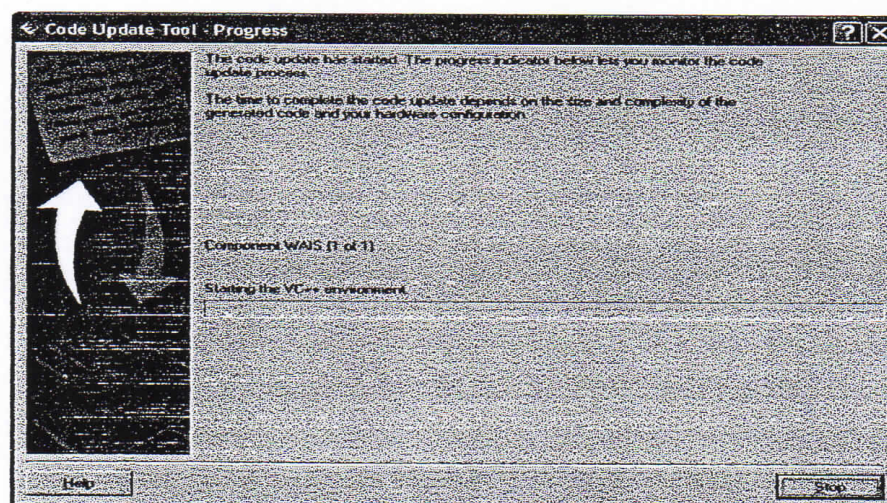
5. Tampil jendela 'Code Update Tool – Finish' seperti tampak pada gambar. 9 berikut:





**Gambar. 9. Jendela ‘Code Update Tool – Finish’**

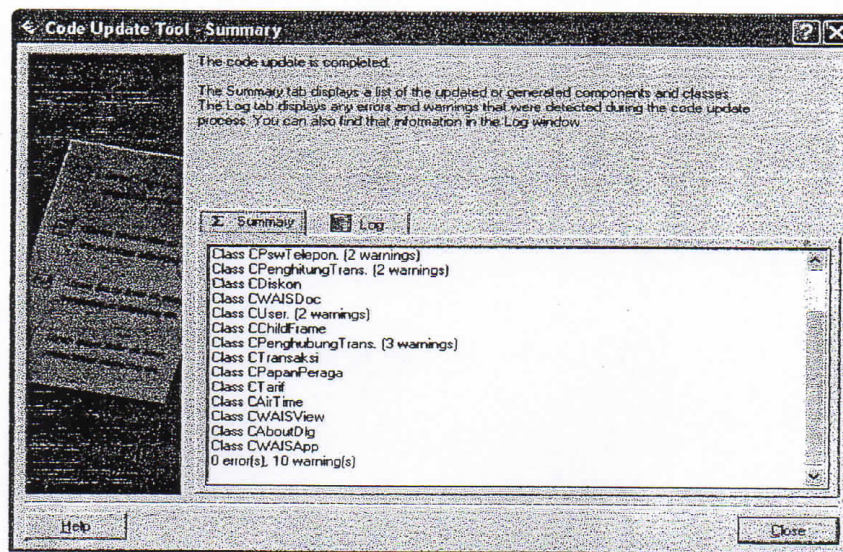
Klik tombol 'Finish' maka sistem akan melakukan proses forward engineering. Untuk melihat jalannya proses tersebut ditampilkan jendela 'Code Update Tool – Progress' seperti tampak pada gambar.10. berikut:



**Gambar. 10. Jendela ‘Code Update Tool – Progress’**

Setelah proses selesai maka akan ditampilkan jendela ‘Code Update Tool – Summary’ seperti yang tampak pada gambar. 11. berikut:





Gambar. 11. Jendela 'Code Update Tool – Summary'

Jendela 'Code Update Tool – Summary' akan menginformasikan error dan warning yang terjadi selama proses forward engineering, misalnya ada atribut yang belum diset tipe datanya maka akan muncul info error dan atribut tersebut tidak akan di-forward ke project aplikasi. Pada tahap ini proses forward engineering telah selesai. Klik tombol 'Close' untuk menutup jendela ini.

## PENUTUP

*Forward Engineering* adalah fitur khusus pada Rational Rose yang membantu pemrogram untuk mengkonversi dari model desain ke bahasa pemrograman.

Hasil utama dari Forward Engineering adalah terbentuknya skema program dari tiap tiap kelas yang terdapat dalam model.

---

**DAFTAR PUSTAKA**

- Booch, Rumbaugh, Jacobson, 1999," *The Unified Modelling Language User Guide* " , Addison Wesley
- Coad, P. and Yourdan,E., 1991, "*Object-Oriented Design*", Prentice Hall
- Davis, A., 1995, " *Object-Oriented Requirements to Object-OrientedDesign: An Easy Transition?*" J. Systems Software, vol 30, pp.151-159
- Anonym, <http://www-pcd.stanford.edu/asd/info/articles>
- Hariyanto, B., 2004, "*Rekayasa Sistem Berorientasi Objek*", Penerbit Informatika Bandung,
- Hermawan, J., 2004, "*Analisa & Desain Pemrograman Berorientasi Obyek dengan UML dan Visual Basic. Net*" , Penerbit Andi Yogyakarta
- Lenti, F.N., 2001, "Penerapan OOA dan OOD Coad Yourdan untuk Analisis dan Perancangan Sistem Informasi Wartel", Puslitbang STMIK AKAKOM
- Pressman R.S., 1997," *Software Engineering A Practitioner's Approach*", Mc Graw-Hill
- Suhendar, A., Gunadi, H., 2002, "*Visual Modelling Menggunakan UML dan Rational Rose* " , Penerbit Informatika Bandung